

# Nulabilidad





1.

## Control de los nulos

“

Si una variable puede ser nula, el compilador nos obligará a manejarla de alguna manera y no nos dejará seguir si no hacemos comprobaciones previas



**2.**

**Nomenclatura**



Como todo es un objeto, todo puede ser nulo. Para ello, simplemente debemos añadir un signo de interrogación al final de su tipo.

*val name: String? = null*



3.

**Operadores**

# Operadores

- ▶ **?.** - Ej: *name?.toString()*

Sólo se ejecutará si la variable es distinta de nula. En caso contrario no hará nada.

- ▶ **?:** - Ej: *name?.toString() ?: ""*

Indicaremos qué hacer si la variable es nula.

- ▶ **!!** - Ej: *name!!.toString()* → ¡Cuidado!

Estamos obligando al compilador que ejecute dicho método aunque sea nula nuestra variable

4.

**Función *let***





Especialmente útiles para los nulos encadenados.

*person?.name?.let { printLog(it) }*



**5.**

**Variables *lateinit***



Variables no nulas pero que no podemos definir en el constructor, ya que por ejemplo la inicializaría un inyector de dependencias

- ¿Cómo sabemos si está inicializada?

Usaremos el método *.isInitialized*